

524-61  
118

188144

# Real-Time Graphic Simulation for Space Telerobotics Applications

E.W. Baumann

McDonnell Douglas Aerospace Information Services Co.  
St. Louis, MO 63166

MP 532 738

**Abstract.** Designing space-based telerobotic systems presents many problems unique to telerobotics and the space environment, but it also shares many common hardware and software design problems with Earth-based industrial robot applications. Such problems include manipulator design and placement, grapple-fixture design, and of course the development of effective and reliable control algorithms.

Since first being applied to industrial robotics just a few years ago, interactive graphic simulation has proven to be a powerful tool for anticipating and solving problems in the design of Earth-based robotic systems and processes. Where similar problems are encountered in the design of space-based robotic mechanisms, the same graphic simulation tools may also be of assistance.

This paper describes the capabilities of PLACE, a commercially available interactive graphic system for the design and simulation of robotic systems and processes. A space-telerobotics application of the system is presented and discussed. Potential future enhancements are described.

is described

## 1. Introduction

As the number and complexity of robot applications increase, the importance of being able to effectively evaluate design and programming alternatives will also increase. While such evaluation can, in most cases, be performed on a trial-and-error basis in the "real world", there are advantages to be gained by first testing those ideas in a simulated environment using computer graphics. The major advantages are:

- \* The time and materials spent physically prototyping alternative robotic systems is reduced or eliminated.
- \* The possibility of inflicting physical harm to personnel or equipment in the event of a programming error is reduced.
- \* Characteristics of the space environment that cannot be physically reproduced on Earth may be amenable to computer simulation.

The major disadvantage of using computer simulations to develop robotic systems and processes is that simulations are never perfect representations of what will happen in the real world. The user must therefore be careful to understand which aspects of the real world behavior are important to his application and how well they are reproduced by the simulation.

The following portions of this paper discuss some areas of robotics in which graphic simulation tools can be of value and describe several products produced by McDonnell Douglas for this purpose.

## 2. Conceptual Design Using Graphic Simulation

Before detailed design work can begin on a new robot application, it is first necessary to develop a general concept of the system and processes that will be required to achieve the specified goal within a particular environment. This conceptual design is useful not only as the initial step in the top-down design of a new robotic system, but also as a means to effectively describe your concepts to others. It is generally easier to get an idea across by viewing the simulated system in action than by reading pages of text and static drawings.

PLACE (Positioner Layout and Cell Evaluator) was the first in a series of McDonnell Douglas Robotics Software Products. It executes on DEC VAX 11/780, 11/750, and Micro-VAX computers using an Evans & Sutherland PS300 Computer Graphics System. The PLACE software is designed to graphically create, analyze, and modify robotic "work-cell" descriptions. A "work-cell" description is a collection of CAD-generated geometry representing the components of a robot-based manufacturing system or "work-cell". These components include robots, end-effectors, fixtures, NC machines, raw material, completed parts, and miscellaneous tooling. The designer has the option of creating an original cell description or using McDonnell Douglas supplied robots or work-cells found in the library of cell description files. These files contain models of many commercially available robots, as well as cells for a variety of robot applications.

PLACE includes the following features:

- \* Kinematic equations to simulate the motion of over 100 industrial robots.
- \* Continuous readout of joint angle data during robot motion.
- \* 3-D graphics for manipulation, motion specification, and visual collision detection (automatic collision detection is also available).
- \* Interactively controlled dynamic 3-D scaling, translating, and rotating of parts, devices, and the entire cell.
- \* Recording of robot motion sequences for playback and analysis.
- \* The ability to define attachment/detachment of parts.
- \* Simulation and programming of device I/O.
- \* Sensor support.
- \* Conditional execution based upon internal computation or device I/O.
- \* Parallel, coordinated device motion.
- \* A user-expandable library of robots and other work-cell components.

The addition of new robots to the library referred to above has been greatly simplified by a software package called "BUILD". BUILD automatically determines the kinematic equations of a robot manipulator from its geometric model, thereby eliminating the need to perform a custom kinematic analysis for each new robot. This makes it easy for the user to test many different manipulators or many variations of the same manipulator in order to approach an optimal design for the task to be performed. Since BUILD is limited to devices having six degrees of freedom or less, the PLACE system includes the ability to define "Compound Devices" comprised of suitably coordinated sub-devices. In this way mechanisms having greater than six coordinated degrees of freedom can be simulated.

### 3. Off-Line Programming

Off-line programming of robotic systems may eventually prove to be the most important application of graphic simulation tools. As robots are required to perform increasingly complex tasks in less structured environments, greater emphasis will be placed on the sensing and logical control aspects of robot programs. One way to generate such programs is to combine motion sequences produced by a system like PLACE with the remainder of a program written in the robot's native language. In this way an off-line program can be created that will already have the robot motion portions largely debugged.

McDonnell Douglas has produced a system called "COMMAND" that provides a set of translators for generating off-line programs from motion sequences created using PLACE. The translators also process instructions entered in the robot's native language. These instructions can include references to the motion sequences defined previously in PLACE. Translator output consists of a robot source program and an object code data file. This data file can be automatically written to tape or diskette as required for loading into the robot controller.

### 4. Space Telerobotics Example

Specific questions in the realm of space telerobotics that could be resolved, at least in part, by someone using PLACE and BUILD include:

- \* Can an "off-the-shelf" industrial robot be used for a space-based application? If not, then can a slightly modified version be used?
- \* Can a modular, reconfigurable manipulator capable of supporting a wide range of assembly and repair tasks be designed?
- \* Can a general-purpose gripper for space-based assembly and repair tasks be designed?
- \* Where should the base of a manipulator on the space station be located in order to perform a cooperative task with the shuttle manipulator?
- \* Where should the manipulator(s) be located on a teleoperated maneuvering vehicle (TMV)?
- \* How should the tool-bay of a TMV be organized? Can the tools be reached by the manipulator(s)?
- \* Where should the cameras be located? Where should they be looking during a particular stage of the process?
- \* If a manipulator needs to move a payload between two points, what paths are collision free and do not cause joint limit errors?

To illustrate and assess the capabilities of PLACE and BUILD as applied to the space-telerobotics domain, an example telerobotics scenario was developed and used as the basis for a PLACE demonstration. The basic scenario is shown in figures 1 through 7, plotted directly from the PLACE display. Here are descriptions of each figure and a few comments regarding the simulation at that point:

**Figure 1 - Shuttle Arriving at the Space Station**

All major movable components of the station are modelled as devices. This includes the solar panels, the large radio dish antenna, and the waste-heat radiator panels (below the radio dish). The large box-like structure located between the solar panels on the center truss is a hangar in which astronauts can perform satellite repair work. The primary goal at this point is to move the shuttle safely into a good position for transferring the laboratory module from the shuttle to the station manipulator. It is important that the position chosen not cause joint errors in either manipulator or require repositioning of the shuttle during the module transfer. Locations satisfying these constraints can be readily found by having the station and shuttle manipulators "track" the module as you use the simulator's control dials to change the shuttle vehicle's position while monitoring the manipulator joint displays.

**Figure 2 - Laboratory Module Being Transferred From the Shuttle Manipulator to the Station Manipulator.**

The main problem encountered here was in determining the locations for the two grapple fixtures on the module. The center grapple location worked well for the shuttle manipulator. Having the station manipulator grab the module on top and then moving the station manipulator's platform vertically to insert the module required minimal motion of the station manipulator.

**Figure 3 - Teleoperated Maneuvering Vehicle (TMV) Preparing to Capture a Satellite.**

Each of the TMV's main arms has six degrees of freedom. Each finger has five degrees of freedom. The arms and fingers can be controlled independently or as a single "Coordinated Motion Compound Device". The vehicle has two cameras, one located on the left boom and one on the right. Two disk shaped communication antennas are located behind and slightly below the camera booms. The fingers on the TMV's right hand are brought to a point for insertion into the recessed nozzle of the satellite. The fingers of the left hand are opened to form a flat surface to push against the opposite end of the satellite.

**Figure 4 - TMV Transferring Satellite to Space Station Hangar.**

Here the TMV is getting into position to deposit the satellite into the hangar. It's approach from "below" (between the hangar and the radiators) requires that the station's antenna be moved out of the way to avoid a possible collision. It may have been better to enter the hangar from above (where the two trusses meet) but in that case there might still be a need to reposition the solar panels to reduce the chance of collision.

**Figure 5 - Station Climbing, Observing, and Repair (SCORP) Vehicle Attached to Station Manipulator.**

By replacing the three-fingered hands with cylindrical grippers suitable for grasping space station struts, replacing the fixed cameras on booms with movable cameras on "eyestalks", and by adding a "tail" capable of securely latching onto any portion of the station's truss structure, the TMV can be converted into a vehicle capable of climbing on, inspecting, and repairing the space station. It is shown here attached to the station manipulator prior to being placed on the truss structure. The SCORP has no engines and therefore must always be attached to the station in some way. A parts bay is located inside the SCORP's "chest" below its left arm. A tool bay is located below its right arm.

**Figure 6 - SCORP Climbing on Station While Performing Visual Inspection**

Climbing is accomplished by declaring the arms and body of the vehicle to be a "Coordinated Motion Device", thereby forcing them to begin and end their motion simultaneously. The requirement that one hand continue to grasp the station while the body and arms are being repositioned is indicated to the PLACE system by temporarily declaring the corresponding arm to be a "Dependent Motion Device". The system then asks the user to specify the spatial relationship (in this case between hand and strut) that must be maintained during the execution of this climbing step. Once all of the goal positions and dependencies are defined, the simulation can proceed. The same approach can be used to simulate walking.

Figure 7 - SCORP Anchored to Station While Welding Reinforcement Strut to Space Station Box-Truss Structure.

When a repair job requires the use of both arms, the SCORP's "tail" can be used to grab the station structure, thereby freeing the arms while still providing substantial local mobility for the vehicle. The grapple position shown in this figure actually causes the tail-arm to exceed two joint limits in its wrist and should be changed. Exceeding joint limits in this way is a common error that would be difficult and time consuming to identify without the use of a simulation system like PLACE.

#### 5. Future Directions

The primary emphasis of these "first generation" graphic simulation systems has been on providing user-friendly methods for specifying robot motion, and then accurately portraying the programmed motion. There is no doubt that additional improvements can, and will, be made in these areas.

As more complex robot applications appear, greater attention will be paid to simulating sensor-based robot behavior and in keeping track of accumulated position and force errors that may lead to failure. Automatic generation of "sensory expectations" for the robot's sensory systems will also be necessary if a complete off-line programming environment is to be realized.

So far these systems have only acted as providers of relatively raw information to a human decision maker. In the not-too-distant future we may see systems having the ability to analyze their own simulation results and give advice to the human user during a cell design or programming session. Perhaps planning some parts of the process, such as finding collision-free manipulator trajectories, will gradually be turned over to the system, with the human user acting increasingly as "supervisor" rather than programmer. Ultimately there will cease to be a need for graphic output from the robot simulation and planning system, except to serve as a window into the machine's planning process as it determines how to accomplish to goals we have set before it.

#### 6. Acknowledgments

I would like to thank Nicholas Denissen for his assistance in creating several of the Unigraphics CAD models used in the space telerobotics example.

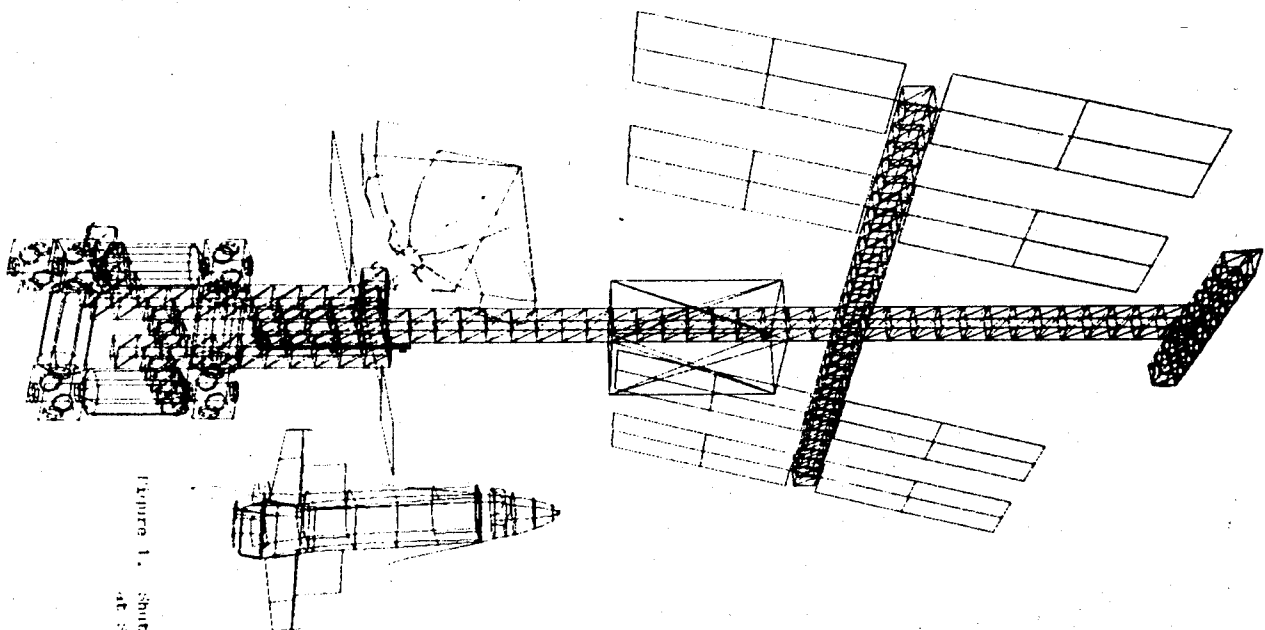


Figure 1. Shuttle Arriving  
at Space Station

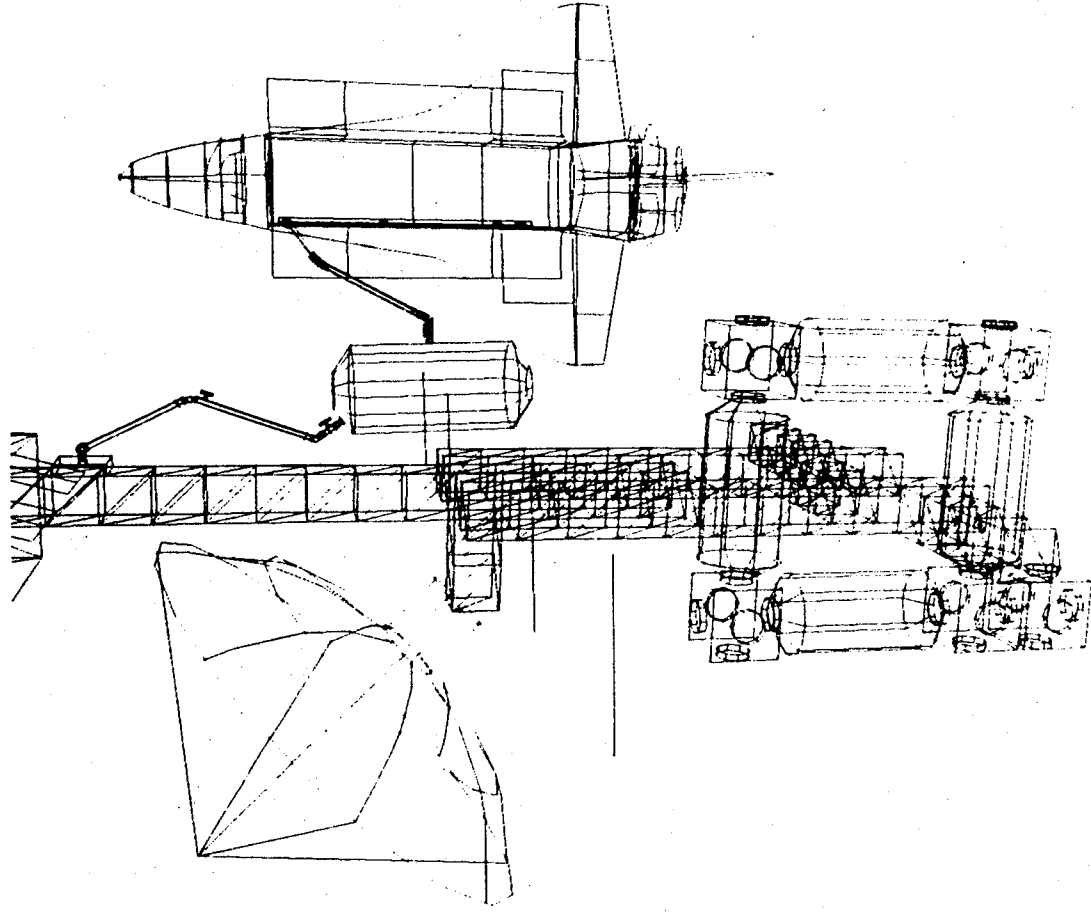


Figure 7. Laboratory Module Being Transferred From Shuttle Manipulator to Station Manipulator

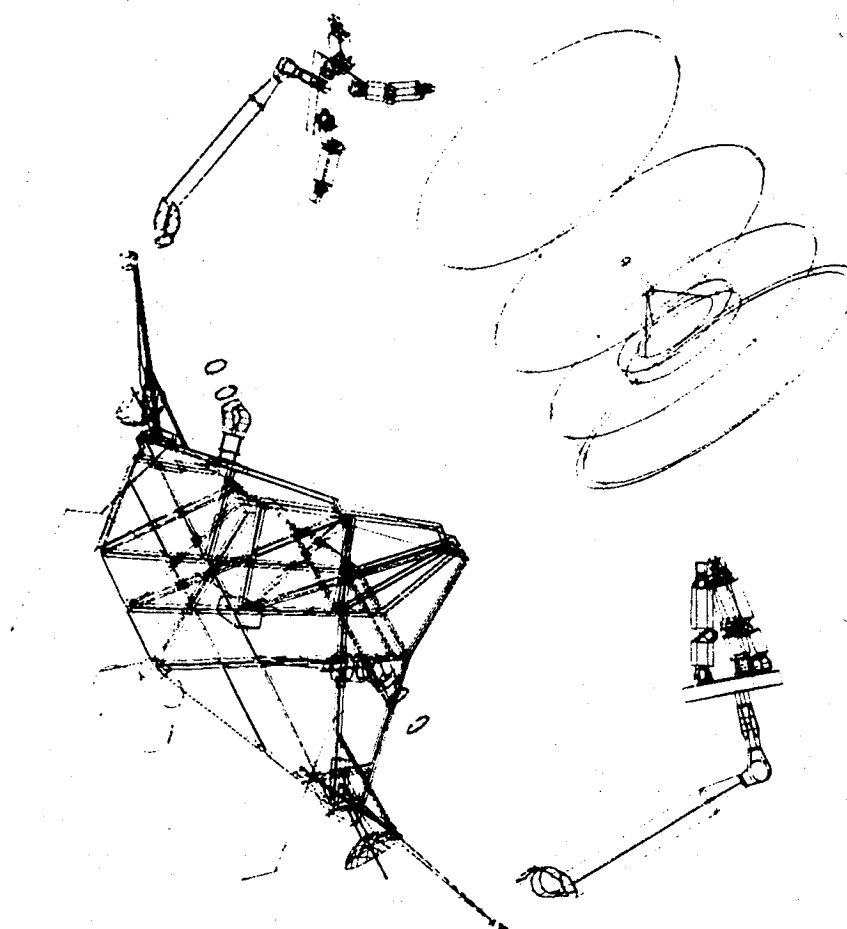


Figure 3. Teleoperated Maneuvering Vehicle (TMV)  
Preparing to Capture a Satellite

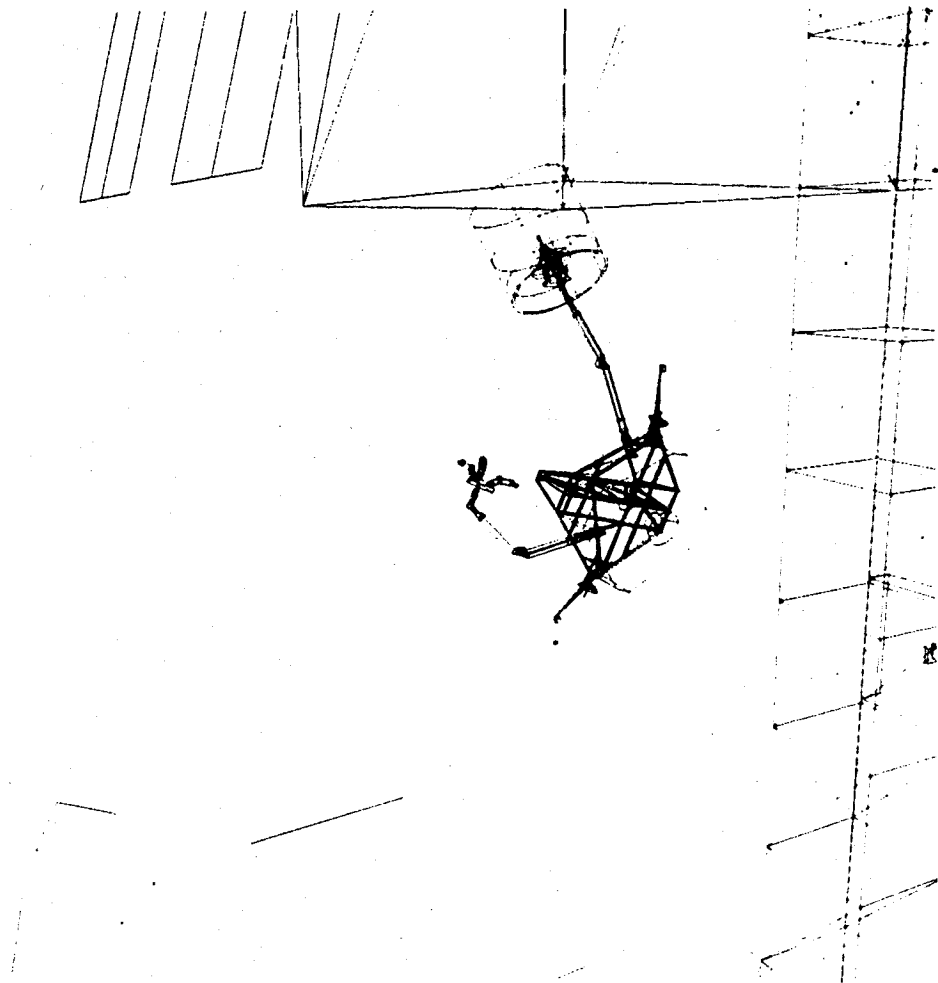


Figure 4. TMV Transferring Satellite to Space Station Bangar

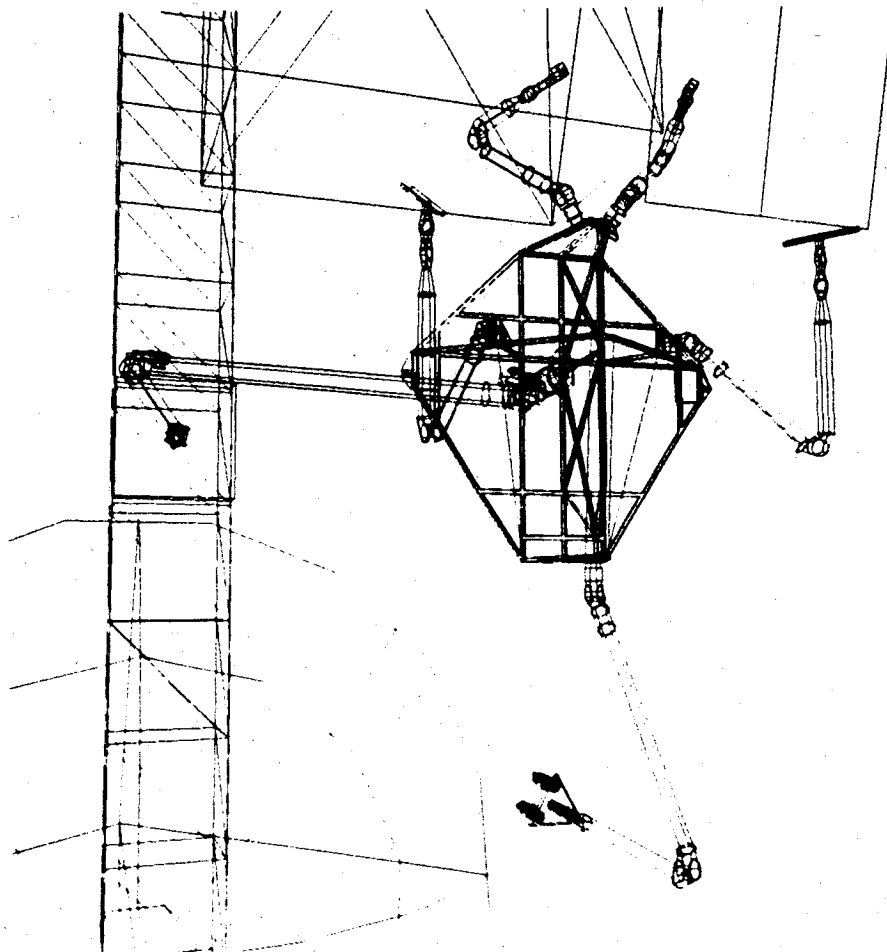


Figure 5. Station C. (M), Observing, and Repair Vehicle (SCORP)  
Attached to Station Manipulator

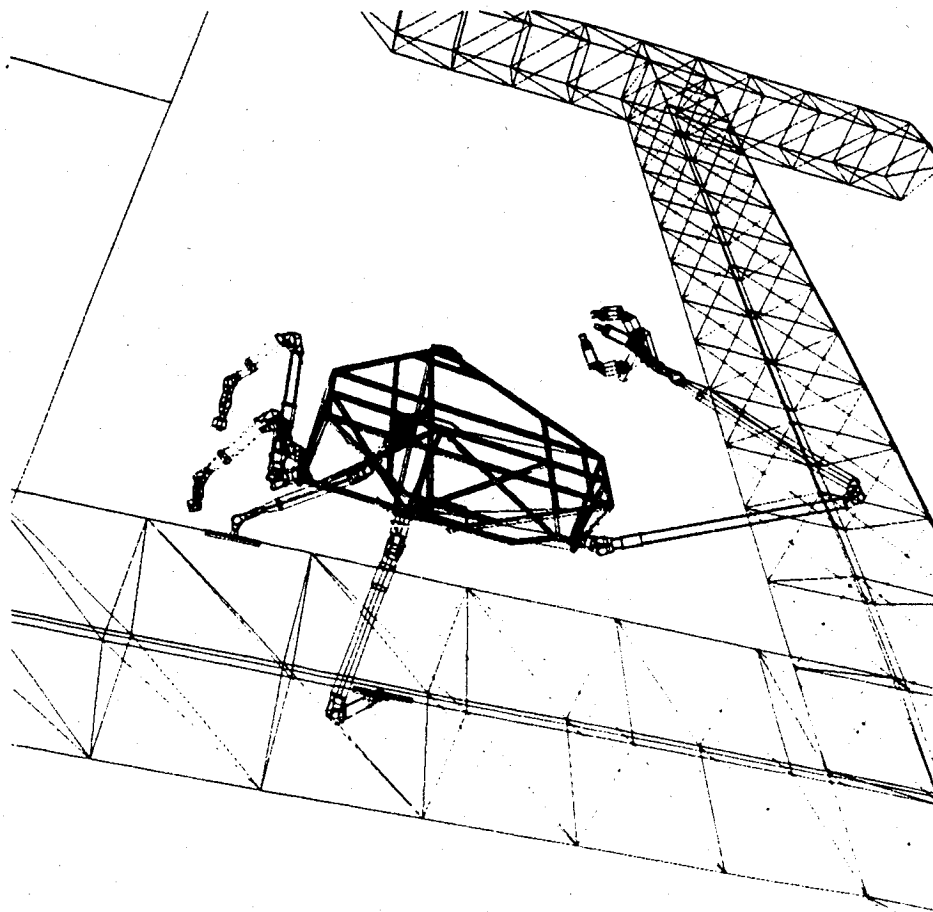


Figure 6. SCORP Climbing on Station While Performing Visual Inspection

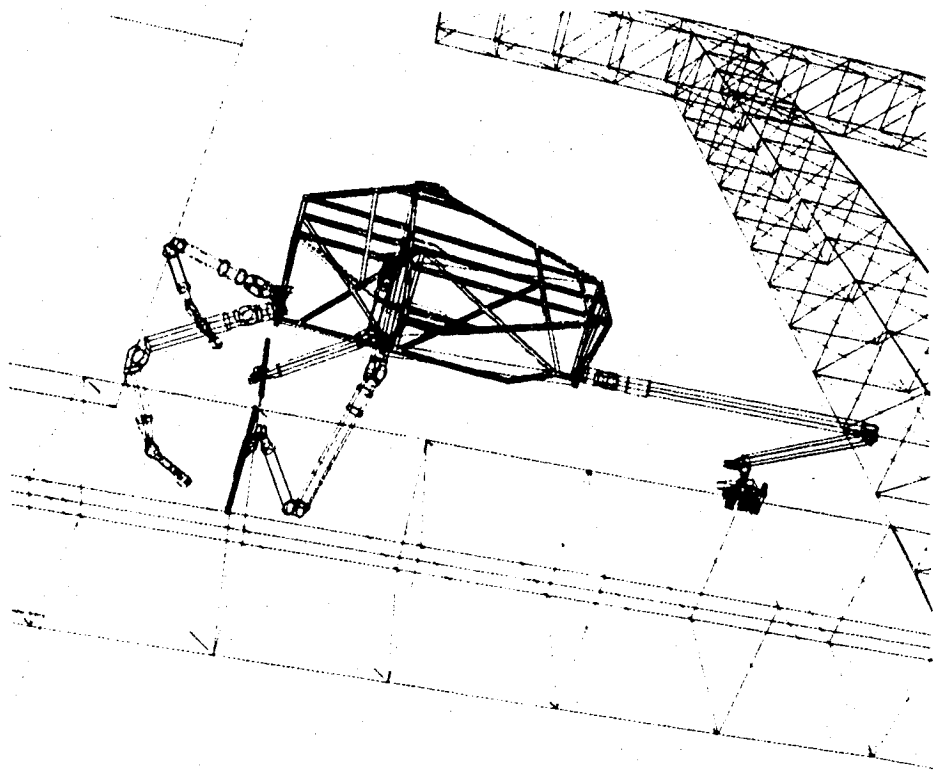


Figure 7. SCORP Anchored to Station While Welding Reinforcement  
Strut to Space Station Box-Truss Structure